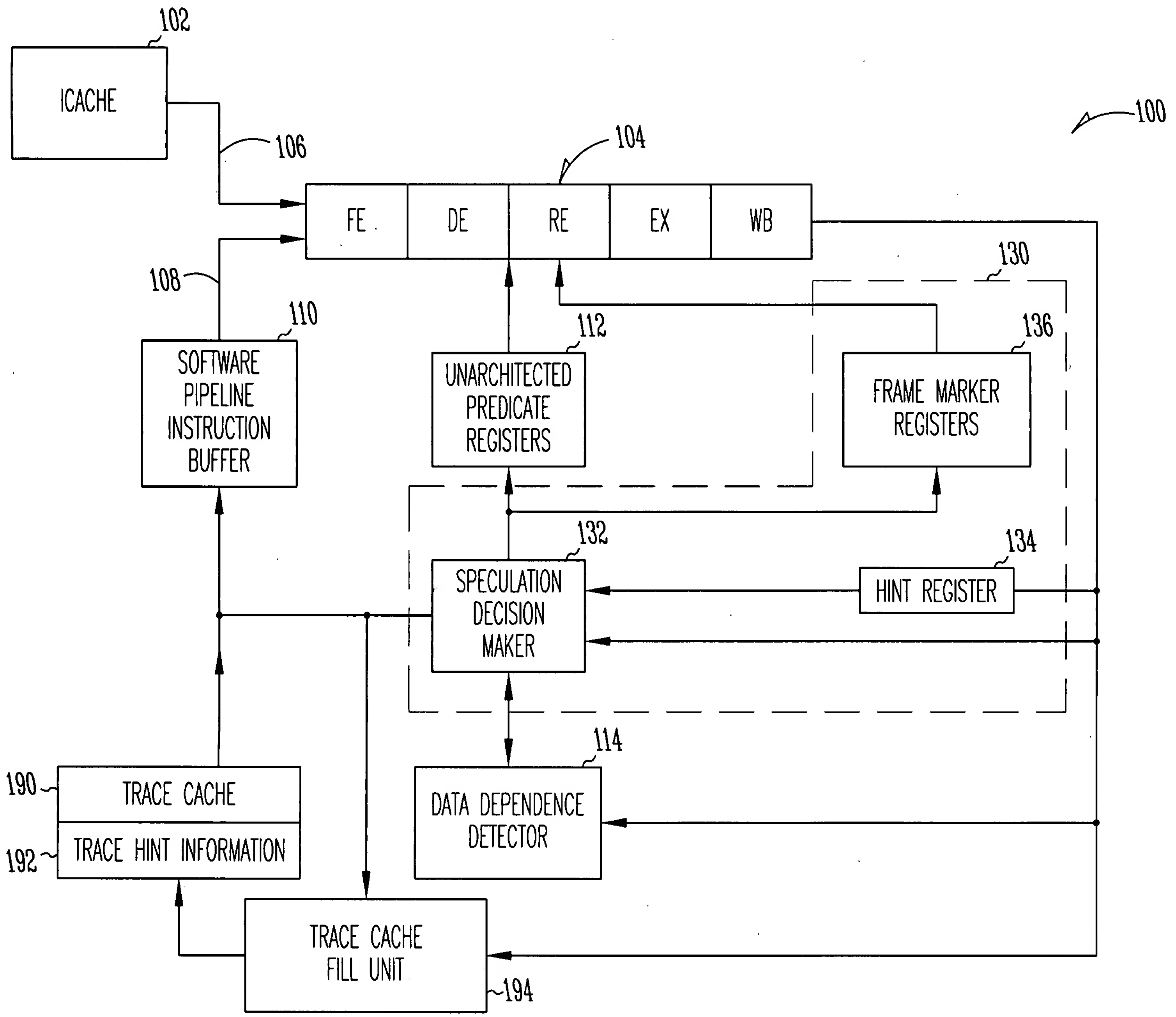
100
102
ICACHE
106
104
FE
DE
RE
EX
WB
108
110
SOFTWARE PIPELINE INSTRUCTION BUFFER
112
UNARCHITECTED PREDICATE REGISTERS
130
136
FRAME MARKER REGISTERS
132
SPECULATION DECISION MAKER
134
HINT REGISTER
114
DATA DEPENDENCE DETECTOR
190
TRACE CACHE
192
TRACE HINT INFORMATION
TRACE CACHE FILL UNIT
194

*Fig. 1*

```
for (i=o; i<n; i++)
     y[i] = x[i] + 1;
```

*Fig.2A*

```
// Initialization
 mov pr.rot  = 0             // Clear all rotating predicate registers
 cmp.eq p16,p0 = r0,r0       // Set p16=1
 mov ar.lc = 4               // Set loop counter to n-1
 mov ar.ec = 3               // Set epilog counter to 3

// loop
loop:
    (p16) ldl r32 = [r12],1     // Stage 1:  load x
    (p17) add r34  = 1,r33      // Stage 2:  y=x+1
    (p18) stl [r13] = r35,1     // Stage 3:  store y
    br.ctop.sptk.few loop       // Branch back
```

*Fig.2B*

```
    // loop
    loop:
    (p16) ldl r32 = [r12],1      // Stage 1:  load x
310 (p17) add r34  = 1,r33      // Stage 2:  y=x+1
    (p18) stl [r13] = r35,1     // Stage 3:  store y

    // loop
    (p63) ldl r127 = [r12],1     // Stage 1:  load x
320 (p16) add r33  = 1,r32      // Stage 2:  y=x+1
    (p17) stl [r13] = r34,1     // Stage 3:  store y

    // loop
    (p62) ldl r126 = [r12],1     // Stage 1:  load x
330 (p63) add r32  = 1,r127     // Stage 2:  y=x+1
    (p16) stl [r13] = r33,1     // Stage 3:  store y

    // loop
    (p61) ldl r125 = [r12],1     // Stage 1:  load x
340 (p62) add r31  = 1,r126     // Stage 2:  y=x+1
    (p63) stl [r13] = r32,1     // Stage 3:  store y
```

*Fig. 3*

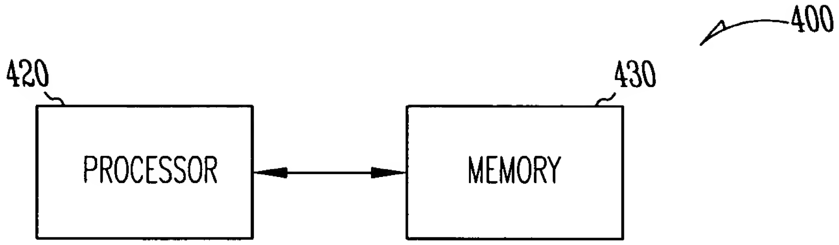

*Fig. 4*